Nama: Raditya Sebastian

NIM:1306620042

Resume 4

**Image Convoultion**

Dalam pemrosesan gambar, konvolusi adalah proses mengubah gambar dengan menerapkan kernel pada setiap piksel dan tetangga lokalnya di seluruh gambar. Kernel adalah matriks nilai yang ukuran dan nilainya menentukan efek transformasi dari proses konvolusi.

Proses Konvolusi melibatkan langkah-langkah ini.

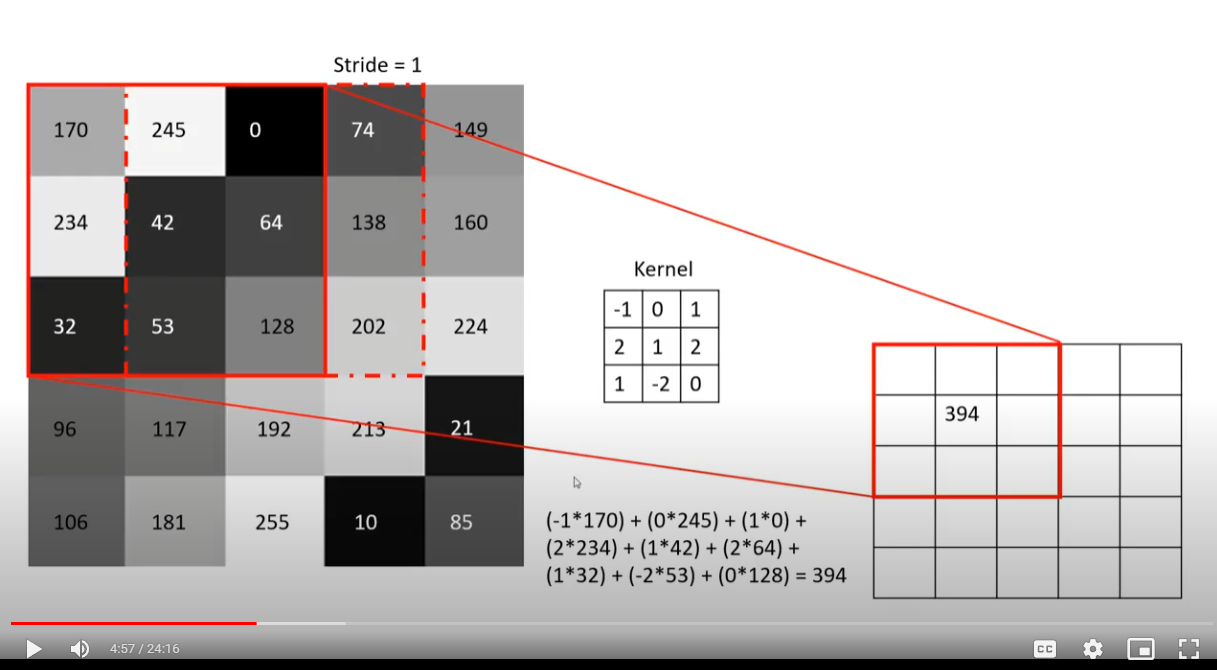
(1) Ini menempatkan Kernel Matrix di atas setiap piksel gambar (memastikan bahwa Kernel penuh ada di dalam gambar), mengalikan setiap nilai Kernel dengan piksel yang sesuai dengannya.

(2) Kemudian, menjumlahkan nilai yang dikalikan yang dihasilkan dan mengembalikan nilai yang dihasilkan sebagai nilai baru dari piksel tengah.

(3) Proses ini diulangi di seluruh gambar.

Contoh:

Sebuah pixel seperti dibawah ingin di convulusi dengan kernel maka prosesnya akan seperti ini



Stride itu adalah melakukan proses ini ke posisi selanjutnya sebanyak n, dan dari contoh diatas adalah sebanyak 1 maka matrix perhitunganya bergerak satu pixel sehingga menghasilkan matrix baru.

Apa yang terjadi jika tidak cukup saat mengkonvulusikanya,itu bisa dijawab dengan menggunakan metode padding, jadi padding itu menambahkan kaya sebuah layer matrix lagi yang melingkari pixelnya sehingga bentuknya akan selalu 3x 3 atau 6x6, nilai paddingnya adalah 0.

Dari sini ada beberapa data type yang bisa membantu untuk membuat image processing ini.

| **Data type** | **Range** |
| --- | --- |
| uint8 | 0 to 255 |
| uint16 | 0 to 65535 |
| uint32 | 0 to 232 - 1 |
| float | -1 to 1 or 0 to 1 |
| int8 | -128 to 127 |
| int16 | -32768 to 32767 |
| int32 | -231 to 231 - 1 |

Data type di atas bisa diakses oleh library skimage

Dan ini semua bisa diterapkan pada kode

Contoh:

import numpy as np  
from skimage import io, img\_as\_float  
import cv2  
img = img\_as\_float(io.imread("Photos/123.jpg",as\_gray=True))  
print(img)  
#Normalize Box filter  
#taking averages of all pixel of kernel area and replace the center element with whatever that average is  
kernel = np.ones((5,5),np.float32)/25  
print(kernel)  
#we can also use gaussian kernel  
gaussian\_kernel = np.array([[1/16,1/8,1/16],[1/8,1/4,1/8],[1/16,1/8,1/16]])  
print(gaussian\_kernel)  
#now we can apply all that stuff to the image  
#there couple way to do it  
#open cv route  
#applying Filter2d  
conv\_using\_cv2=cv2.filter2D(img,-1,kernel,borderType=cv2.BORDER\_CONSTANT)#gambar,depth(making sure the output same as the depth),kernel,border  
conv\_using\_cv2\_gauss=cv2.filter2D(img,-1,gaussian\_kernel,borderType=cv2.BORDER\_CONSTANT)  
  
print(conv\_using\_cv2)  
cv2.imshow('Original image',img)  
cv2.imshow('Hasil Convolusi',conv\_using\_cv2)  
cv2.imshow('Hasil Convolusi\_gauss',conv\_using\_cv2\_gauss)  
cv2.waitKey(0)

Dari gambar yang dihasilkan akan menghasilkan blur yang terkadang merusak gambarnya untuk mengurangkan blur itu kurangkanlah nilai kernelnya.

Ada juga cara menghaluskan gambarnya dengan menggunakan rata ratanya. Dan cara ini sudah disiapkan pada opencv yaitu:

Smoothing Images

Contohnya:

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

img = [cv.imread](https://docs.opencv.org/4.x/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56)('opencv\_logo.png')

kernel = np.ones((5,5),np.float32)/25

dst = [cv.filter2D](https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html#ga27c049795ce870216ddfb366086b5a04)(img,-1,kernel)

plt.subplot(121),plt.imshow(img),plt.title('Original')

plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(dst),plt.title('Averaging')

plt.xticks([]), plt.yticks([])

plt.show()

**Blur dan Filter**

**A.Blur**

Pengaburan adalah untuk membuat sesuatu yang kurang jelas atau berbeda. Ini dapat ditafsirkan secara luas dalam konteks analisis gambar - apa pun yang mengurangi atau mendistorsi detail gambar mungkin berlaku. Menerapkan low pass filter, yang menghilangkan detail yang terjadi pada frekuensi spasial tinggi, dianggap sebagai efek blur. Blur Gaussian adalah filter yang menggunakan kernel Gaussian.

Contoh:

import cv2 as cv  
img = cv.imread('Photos/123.jpg')  
#blur(blur can be use to unblur a image or blur a image)  
blur= cv.GaussianBlur(img,(3,3),cv.BORDER\_DEFAULT)#core image,kernel size(window size[has to bee odd number

**B. Filter**

Filter dengan lubang kecil hanya memungkinkan benda-benda kecil masuk, meninggalkan benda-benda yang lebih besar di belakang. Ini adalah analogi yang bagus untuk filter gambar. Filter high-pass akan mempertahankan detail yang lebih kecil dalam gambar, memfilter yang lebih besar. Filter low-pass mempertahankan fitur yang lebih besar, analog dengan apa yang ditinggalkan oleh mesh filter fisik. High- dan low-pass, di sini, mengacu pada frekuensi spasial tinggi dan rendah dalam gambar. Detail yang terkait dengan frekuensi spasial tinggi kecil, banyak dari fitur ini akan cocok di seluruh gambar. Fitur yang terkait dengan frekuensi spasial rendah berukuran besar - mungkin beberapa fitur besar per gambar.

Contohnya adalah Edge Cascading:

#Edge Cascading  
canny = cv.Canny(img,125,175)#image,threshold value  
cv.imshow('Canny-edge',canny)